**1. INTRODUCTION**

**1.1 PURPOSE**

The purpose of this document is to build an android application which digitalises the European accident report.

**1.2 DOCUMENT CONVENTIONS**

This document uses the following conventions.

|  |  |
| --- | --- |
| DB | Database |
| DDB | Distributed Database |
| ER | Entity Relationship |

**1.3 INTENDED AUDIENCE AND READING SUGGESTIONS**

This project is a prototype for the accident reporting application. The application gets build for college programming project. Its intended use is to asses the requirements for said software and therefore define a state of done.

**1.4 PROJECT SCOPE**

The purpose of the accident reporting application is to digitalize the process of accident reporting via an android app. Replacing the paper sheet used for reporting traffic accidents is the ultimate goal of this app. Saving user information and the reporting of accidents is done via REST API build on Loopback 4. Storing user information is done on a Postgres Database. Videos and Photos are stored separately on a server.

**1.5 REFERENCES**

* <https://krazytech.com/projects>
* Fundamentals of database systems by ramez elmarsi and shamkant b.navathe

**2. OVERALL DESCRIPTION**

**2.1 PRODUCT PERSPECTIVE**

A database system stores the following informations.:

* **Vehicles details:**  
  Stores information’s on the vehicle which is part of the accident.
* **Policyholder description:**  
  Includes the Name, Address and email address
* **Insurance description:**  
  Links the policy holder with his insurance company. The insurance information include address of the insurance company and information on the agency providing the policy.

**Addresses description:**  
Stores the addresses of Policy holders, insurance companys, etc.

**Insurance description:**  
Driver information, are used when the driver is different than the policy holder of the car.

**Insurance description:**  
The circumstance table is a list with possible accident circumstances.

**2.2 PRODUCT FEATURES**

The major features of airline database system as shown in below [**entity–relationship model**](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model) (**ER model**)

The diagram shows the layout of the applications database system – entity–relationship model

**2.3 USER CLASS and CHARACTERISTICS**

The users of the system should be able to fill reports of their accidents and include pictures. The report should get send by the system via e-mail. Reports should get send to the selected insurance company and the people who are part of the accident. Pictures and videos that were taken from the accident should get send to each user and the selected insurance company. The application should guide the user through the process of creating a report. If a person is injured, the application should guide the user on applying first aid.

The user should have the following functionalities:

* Create report.
* Take pictures/videos of the accident.
* Receive the report via mail.

Creating the report consists of the following steps:

1. The user wants to create a claim.
2. The user selects the number of people who are part of the accident.
3. The user selects number of witnesses.
4. The user enter information of the policy holder.
5. The user adds information of his vehicle.
6. The user information of the driver, if it differs from the policy holder.
7. The user selects insurance company and agency of the policy holder.
8. The user selects his circumstances of the accident.
9. Optional: Take photos.
10. The user enters his description of the accident.
11. Repeat steps 4-10 for each person that is part of the accident.
12. The witness enters his personal details.
13. The witness describes his version of the accident.
14. Repeat step 12&13 for each witness.
15. Optional: Take pictures.
16. Show report on device.
17. Send report to the insurance company, the people that are part of the accident and the witnesses.

Photos can be taken before the claim gets created as an optional step.

**2.4 OPERATING ENVIRONMENT**

Operating environment for the applications

* Postgresql Database
* REST API build with Loopback 4.(Typescript)
* Android application (Java)

**2.5 DESIGN and IMPLEMENTATION CONSTRAINTS**

1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications
3. How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.
4. Implement the database at least using a centralized database management system.

**2.6 ASSUMPTION DEPENDENCIES**

Let us assume the application is actually in use. We would need access to each insurance company’s API. We assume that the insurance companys use some kind of landing zone and REST API for that case. Therefore we deliver our data in the JSON format with a connected folder, containing the pictures/videos.

**3. SYSTEM FEATURES**

**Features**

* E-mail

The application should be able to send E-Mails to the people who are part of the accident and the witnesses.

* Speech to text converter

When one of the users creates his statement for his version of the accident, a speech to text converter is used to do that.

* Insurance API

When creating the application it is not clear how the Insurance companys API is build. We assume that some kind of REST API is used. Therefore the report information are delivered via JSON file format. The JSON file could be put into REST API. The pictures/videos should be delivered over some kind of landing zone. This landing zone consists of some kind of file storage. The folder/ZIP containing the pictures/videos should be named with the claim number.

**CLIENT/SERVER SYSTEM**

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (Android application), and the backend (Loopback 4 REST API, Postgresql).

The android application sends the information needed for the report to the database via REST API. This enables the application to be independent from the database design.

**4. EXTERNAL INTERFACE REQUIREMENTS**

**4.1 USER INTERFACES**

* Mobile application build with JAVA
* Back-end build with Loopback 4 and Postgresql

**4.2 HARDWARE INTERFACES**

* Android

**4.3 SOFTWARE INTERFACES**

Following are the software used for the accident reporting application.

|  |  |
| --- | --- |
| **Software used** | **Description** |
| Operating system | We are using android as a platform for the mobile application. |
| Database | To save the report information, we use a postgresql database. |
| Java | To implement the project we have chosen to use Java. |

**4.4 COMMUNICATION INTERFACES**

This project supports all types of web browsers. We are using simple electronic forms for the reservation forms, ticket booking etc.

**5. NONFUNCTIONAL REQUIREMENTS**

**5.1 SAFETY REQUIREMENTS**

In order to ensure that the backend is running at all time the backend is deployed on docker. If the container shuts down, or its connection possibility is interfered, it should automatically redeploy. If the server shuts down the container should redeploy on a different server.

**5.2 SECURITY REQUIREMENTS**

Loopback 4 is used as a measurement against opening the insurers API to the public. Therefore the server should monitor connections to the insurers APIs. The REST API should be deployed inside of an DMZ.